



IFCT091PO. Programación páginas web, JavaScript y PHP

Sku: PD_IFCT091PO

Horas: 65

Formato: HTML

OBJETIVOS

Objetivo general.

- Desarrollar aplicaciones web de comercio electrónico.

Objetivos específicos.

- Comprender la programación del lado del cliente y su papel en la interactividad de las páginas web.
- Identificar y manejar con solvencia los objetos fundamentales en JavaScript, como location, history, document y form.
- Implementar técnicas avanzadas de programación asíncrona utilizando promise, async/await y fetch.
- Utilizar mecanismos de almacenamiento local en el navegador como localStorage y sessionStorage.
- Aplicar herramientas y metodologías de depuración, optimización y buenas prácticas para mejorar el rendimiento de aplicaciones JavaScript.
- Comprender la estructura básica del lenguaje PHP para desarrollar aplicaciones web dinámicas.
- Instalar y configurar correctamente un entorno de trabajo para programación web con PHP.
- Implementar técnicas para recoger, validar y gestionar la información aportada por las personas usuarias mediante formularios.
- Utilizar adecuadamente cookies y sesiones para gestionar estados y mantener información entre páginas.
- Aplicar conceptos de programación orientada a objetos y manejo de excepciones en PHP para escribir código robusto y mantenible.
- Comprender el funcionamiento de las bases de datos relacionales y su aplicación en entornos web.
- Aplicar sentencias SQL para la gestión de datos y su integración con páginas dinámicas.
- Analizar estrategias de modelado, normalización y optimización para diseñar estructuras eficientes.
- Implementar mecanismos de seguridad, respaldo y control de integridad en sistemas de gestión de bases de datos.

- Dominar el uso de sistemas de control de versiones y estrategias de trabajo colaborativo con Git.
- Implementar metodologías ágiles y herramientas de integración continua en proyectos web.
- Configurar y gestionar entornos de despliegue automatizados mediante contenedores y orquestación.
- Aplicar técnicas avanzadas de monitoreo, seguridad y escalabilidad en aplicaciones desplegadas en producción.
- Comprender los principios fundamentales y la estructura de funcionamiento de los frameworks React y Angular en el desarrollo de aplicaciones web.
- Configurar entornos de desarrollo adecuados para proyectos con React y Angular, utilizando herramientas específicas como Create React App y Angular CLI.
- Diseñar y construir interfaces dinámicas mediante el uso de componentes, directivas, enlaces de datos y gestión del estado, aplicando buenas prácticas de organización del código.
- Implementar funcionalidades avanzadas en Angular, tales como el enrutamiento modular, la programación reactiva con RxJS y la verificación de calidad mediante pruebas automatizadas con Jasmine y Karma.
- Instalar y configurar correctamente Node.js, gestionando paquetes mediante npm para optimizar el desarrollo de proyectos.
- Estructurar aplicaciones web basadas en Node.js empleando técnicas profesionales de organización de código y gestión de módulos.
- Desarrollar APIs REST utilizando Express, implementando rutas definidas, middleware eficaz y controladores organizados para crear aplicaciones coherentes.
- Implementar procedimientos de validación de datos y técnicas efectivas de manejo de errores para mejorar la fiabilidad y estabilidad de las APIs.
- Integrar aplicaciones desarrolladas en Node.js con diferentes tipos de bases de datos (SQL y NoSQL) y consumir servicios externos para enriquecer su funcionalidad.

CONTENIDOS

Unidad 1. Programación web del lado del cliente con JavaScript.

1. Programación cliente.

- 1.1. Programación cliente.
- 1.2. Programación de páginas web.
- 1.3. Introducción a JavaScript.
- 1.4. Fundamentos de programación.
- 1.5. Objetos en JavaScript.
- 1.6. Los objetos location e history.
- 1.7. El objeto document.
- 1.8. El objeto form.
- 1.9. Otros objetos.
- 1.10. Programación asíncrona en JavaScript.
- 1.11. Promise y async/await.
- 1.12. Consumo de APIs y Fetch.
- 1.13. Almacenamiento en el navegador: localStorage y sessionStorage.

- 1.14. Depuración y herramientas de desarrollo.
- 1.15. Optimización y buenas prácticas en JavaScript.
- 1.16. Módulos en JavaScript.
- Cuestionario de autoevaluación Unidad 1: 30 minutos.
- Actividad de evaluación Unidad 1: 60 minutos.
- Tiempo total de la unidad: 12 horas.

Unidad 2. Programación web del lado del servidor con PHP.

1. Programación servidor.

- 1.1. Introducción a la programación del lado del servidor con PHP.
- 1.2. Instalación.
- 1.3. Crear un sitio web.
- 1.4. Introducción a PHP.
- 1.5. Arrays y estructuras de control.
- 1.6. Funciones.
- 1.7. Incluir archivos.
- 1.8. Programación orientada a objetos.
- 1.9. Herencia.
- 1.10. Recoger datos del usuario.
- 1.11. Validación de formularios.
- 1.12. Trabajar con cookies.
- 1.13. Sesiones en PHP.
- 1.14. Acceso a archivos.
- 1.15. Tratamiento de errores.
- 1.16. Excepciones.
- Cuestionario de autoevaluación Unidad 2: 30 minutos.
- Actividad de evaluación Unidad 2: 60 minutos.
- Tiempo total de la unidad: 12 horas.

Unidad 3. Fundamentos y gestión avanzada de bases de datos.

1. Base de datos.

- 1.1. Bases de datos.
- 1.2. La base de datos de ejemplo.
- 1.3. El lenguaje SQL.
- 1.4. Acceder a la base de datos.
- 1.5. Buscando más funcionalidad.
- 1.6. Autenticación de los usuarios.
- 1.7. Sitios web seguros.
- 1.8. El proceso de compra.

2. Gestión avanzada de bases de datos.

- 2.1. Modelado de datos y normalización.
- 2.2. Índices y optimización de consultas.
- 2.3. Transacciones y control de concurrencia.
- 2.4. Integridad referencial y restricciones.
- 2.5. Procedimientos almacenados, vistas y triggers.

- 2.6. Copias de seguridad, restauración y administración.
- Cuestionario de autoevaluación Unidad 3: 30 minutos.
- Actividad de evaluación Unidad 3: 60 minutos.
- Tiempo total de la unidad: 11 horas.

Unidad 4. Integración y despliegue de proyectos web.

1. Planificación y control de versiones.

- 1.1. Repositorios y flujos de trabajo con Git.
- 1.2. Manejo de ramas, fusiones y estrategias de branching con Git.

2. Gestión del ciclo de vida del proyecto.

- 2.1. Metodologías ágiles (Scrum y Kanban).
- 2.2. Integración continua (CI) con herramientas como Jenkins o GitLab CI.

3. Despliegue y orquestación.

- 3.1. Contenedores y virtualización (Docker).
- 3.2. Orquestación de contenedores (Kubernetes u otras plataformas).
- 3.3. Despliegue continuo (CD) y configuración de pipelines.

4. Mantenimiento y escalabilidad.

- 4.1. Monitoreo y logging (p. ej., ELK Stack, Prometheus).
- 4.2. Seguridad avanzada (WAF, hardening del sistema).
- 4.3. Estrategias de escalado: optimización del rendimiento y disponibilidad.
- Cuestionario de autoevaluación Unidad 4: 30 minutos.
- Actividad de evaluación Unidad 4: 60 minutos.
- Tiempo total de la unidad: 10 horas.

Unidad 5. Frameworks y librerías de front-end.

1. Fundamentos de React.

- 1.1. Configuración del entorno y creación de un proyecto con Create React App.
- 1.2. Componentes funcionales, props y gestión del state.
- 1.3. Ciclo de vida de los componentes y uso de Hooks.

2. Fundamentos de Angular.

- 2.1. Angular CLI y estructura de proyecto.
- 2.2. Data binding, directivas y componentes.
- 2.3. Servicios e inyección de dependencias.

3. Angular avanzado.

- 3.1. Enrutamiento y organización de módulos.
- 3.2. Observables y programación reactiva con RxJS.
- 3.3. Testing con Jasmine y Karma.
- Cuestionario de autoevaluación Unidad 5: 30 minutos.
- Actividad de evaluación Unidad 5: 60 minutos.
- Tiempo total de la unidad: 9,30 horas.

Unidad 6. Node.js y desarrollo de APIs.

1. Configuración y fundamentos de Node.js.

- 1.1. Instalación de Node.js y gestión de paquetes con npm.
- 1.2. Estructura básica de proyectos en Node.js.
- 1.3. Módulos y uso de paquetes externos.

2. Creación de API REST con Express.

- 2.1. Concepto de API REST y enrutado básico.
- 2.2. Middleware, controladores y buenas prácticas de organización.
- 2.3. Validación de datos y manejo de errores.

3. Integración con bases de datos y servicios externos.

- 3.1. Conexión con bases de datos NoSQL (MongoDB) y SQL (MySQL, PostgreSQL).
- 3.2. Uso de ORM/ODM (Sequelize, Mongoose) para simplificar el acceso a datos.
- 3.3. Consumo de APIs y servicios de terceros desde Node.js.
- Cuestionario de autoevaluación Unidad 6: 30 minutos.
- Actividad de evaluación Unidad 6: 60 minutos.
- Tiempo total de la unidad.